National Defence Défense
Defence nationale

APPLICATION OF AUTOREGRESSIVE METHODS TO ADAPTIVE EXCISION OF INTERFERING TONES FROM DIRECT SEQUENCE SPREAD SPECTRUM COMMUNICATIONS SYSTEMS (U)

by

Brian W. Kozminchuk
Communications EW Section
Electronic Warfare Division

DEFENCE RESEARCH ESTABLISHMENT OTTAWA
TECHNICAL NOTE 88-4

PCN
041LK11

March 1988
Ottawa

ABSTRACT

A computer simulation of a BPSK Direct Sequence Spread Spectrum system has been developed to investigate the effects of adaptively filtering stable tone interferers covering twenty percent of the communication band. Preliminary results for the swept tone interfering signal are also presented and shows that the block filter is effective when the tones are stable, but degrades in performance as the tone sweeps quickly across the band. To combat the non-stationary signal environment, either shorter block lengths may be required or the use of recursive algorithms that are based on gradient or least square concepts may have to be examined.

SOMMAIRE

Une simulation informatique du système d'étalement du spectra à séquence directe BPSK a été développée en vue de l'étude des effets de brouilleurs à tonalité avec filtrage adaptatif couvrant 20% de la bande de communications. L'auteur présente les résultats préliminaires dans le cas du signal de tonalité de brouillage avec balayage; on constate que le filtre à blocage est efficace lorsque les tonalités sont stables mais que sa performance diminue lorsque la tonalité varie rapidement sur la bande. Pour remédier aux effets d'un environnement de signaux non stationnaires, on peut réduire la largeur de blocage ou étudier la possibilité du recours à des algorithmes récursifs fondés sur les notions du gradient ou des moindres carrés.

TABLE OF CONTENTS

PAGE

LIST OF FIGURES

LIST OF FIGURES (cont)

LIST OF FIGURES (cont)

1.0 PURPOSE

This report summarizes the work done to date in the area of adaptive excision of interfering signals in Direct Sequence Spread Spectrum systems. It is based on a paper by Ketchum and Proakis [1]. The report fulfills two objectives:

(1) it provides a summary of some adaptive filtering techniques as applied to signal excision, and

(2) presents some preliminary results of work done within the Communications EW section in this area.

2.0 BACKGROUND

In Direct Sequence Spread Spectrum Communication systems, the processing gain may not be sufficient to reduce the effect of strong interferers in the band. Thus, one way to enhance system performance is to incorporate into the receiver an adaptive filter capable of suppressing the interferer sufficiently so that communication can resume at a tolerable bit error rate.

Generally, there are two types of adaptive filter. One is based on processing the data in blocks of size N, and updating the adaptive filter coefficients based on these N samples. The other type is recursive and involves updating the coefficients with every data sample.

The former type works effectively in "stationary" signal environments, i.e., environments in which the interferer is stable for some duration. However, performance may deteriorate if the interferer has some non-stationary characteristics, e.g., a swept frequency. The degree of deterioration would depend on the block size and filter order. The latter type of filter can to some extent handle non-stable types of interferers; but its effectiveness depends on the convergence rate of the filter coefficients relative to the non-stationary characteristics of the interfering signal: e.g., a slow converging filter would have difficulty in tracking a fast swept tone. There are a variety of these types of filters, some of which have faster convergence rates than others.

The primary focus of this report is on the former, i.e., the block processing type of adaptive filter, with passing reference later in the report to the recursive filter.

3.0 BLOCK PROCESSING ADAPTIVE FILTERS

Two types of block processing filter basically exist. They are of the non-parametric and parametric classes.

The non-parametric filters use the data samples $x(nT_s)$, where T_s is the Nyquist interval, of a realization of a stochastic time series $x(t)$ directly to determine the power spectral density (PSD) of the time series, from which an adaptive filter can be designed.

The parametric filters, meanwhile, use the data samples to estimate the parameters of a proposed model for the PSD. Proposed models might be

(i) all pole, also termed Autoregressive (AR) or Maximum Entropy Method (MEM) types
(ii) all zero, also termed Moving Average (MA)
(iii) a combination of both poles and zeroes (ARMA).

3.1 Non-Parametric Filters

An example of the non-parametric approach is one based on the FFT. Here the FFT is used to determine the PSD from the incoming stochastic time series using say,

(i) the Welch technique [6] which averages a series of FFT's on the raw data samples $x(nT_s)$,
(ii) the periodogram technique which involves determining the Fourier transform of the autocorrelation of the time series.

If the PSD is denoted by P(f) and is an even and real function a non-realizable whitening filter can be designed having the transfer function

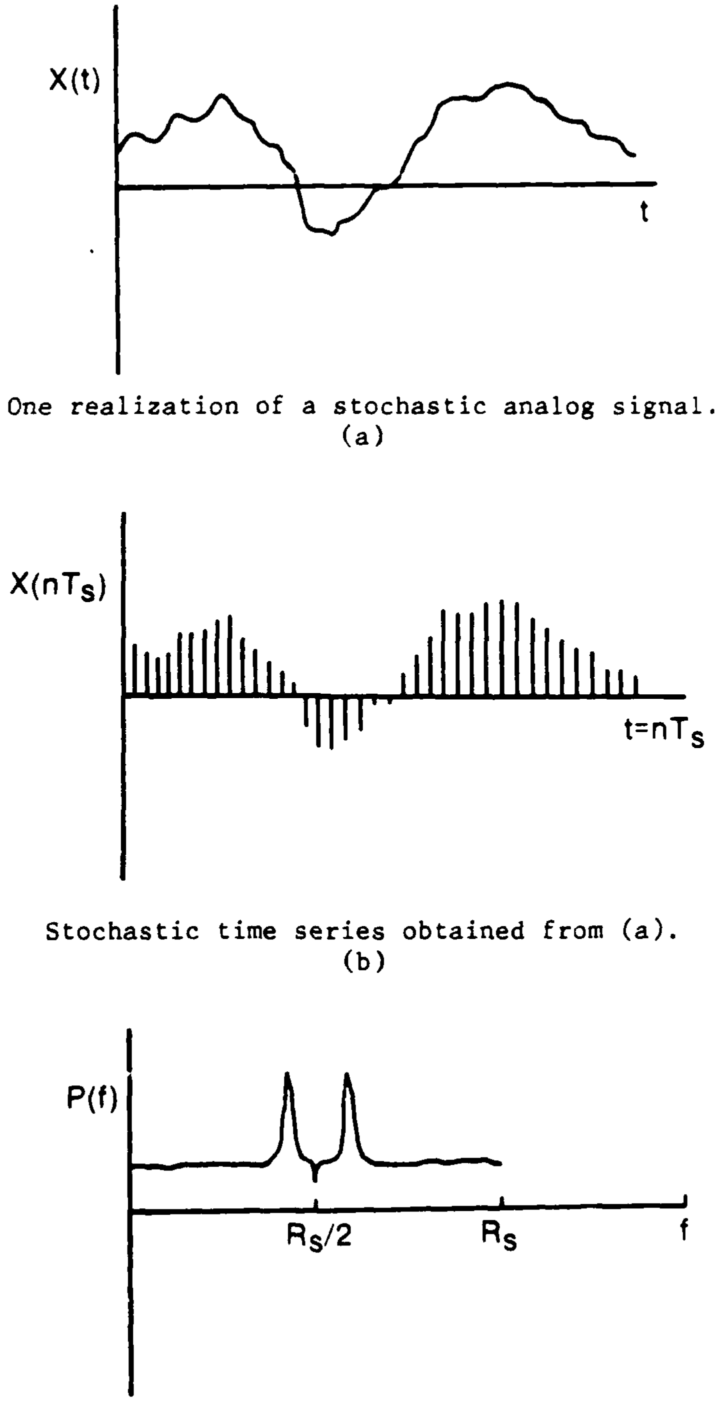
$$H_1(f) = \frac{1}{\sqrt{P(f)}} \qquad (1)$$

Sampling this function M points (where M is odd) around the unit circle, and delaying the signal by (M-1)/2 samples, leads to the realizable linear phase digital filter with transfer function

$$H_1(k) = \frac{1}{P\left[\frac{k}{M} R_s\right]} e^{-j\frac{2\pi}{M}\left[\frac{M-1}{2}\right]k} \qquad (2)$$

where R_s is the sampling rate, equivalent to the chip rate of the pseudo-random sequence. Thus the inverse DFT of H(k) will provide the impulse response h(n), of a transversal filter with M taps and the property that h(n) = h(M-1-n).

Figure 1 shows conceptually how such a filter is created; and Figure 2 indicates how it fits into a system after the signal has been sampled. Finally, Figure 3 contains an example of a 5 tap linear phase filter, with the center tap being the reference because of the (M-1)/2 sample delay through the filter.

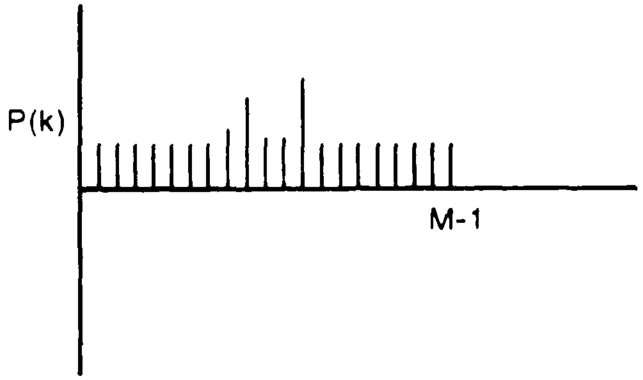


One realization of a stochastic analog signal.
(a)

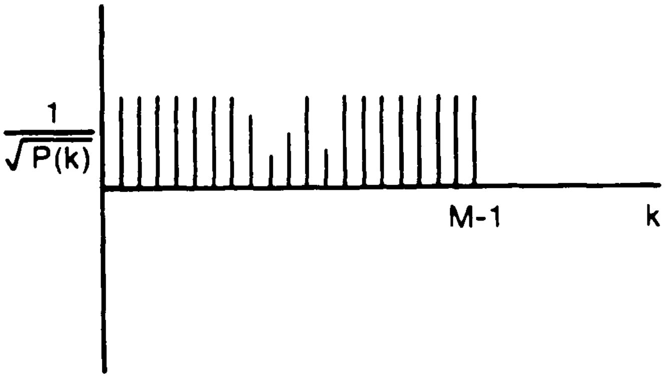
Stochastic time series obtained from (a).
(b)

One-side repeated power spectral density function with folding frequency $R_S/2$. Note that $P(f) = P(R_S - f)$.
(c)

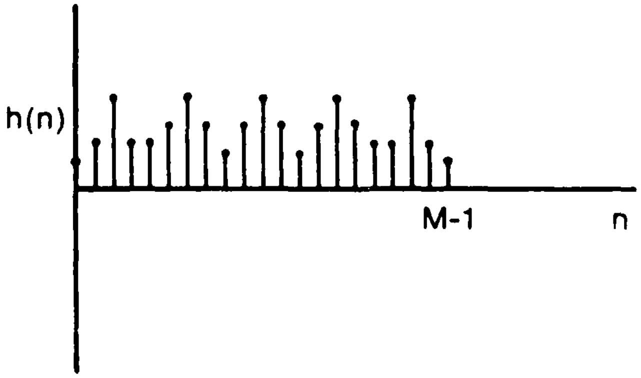
FIGURE 1



Sampled version of power spectral density function.
(d)

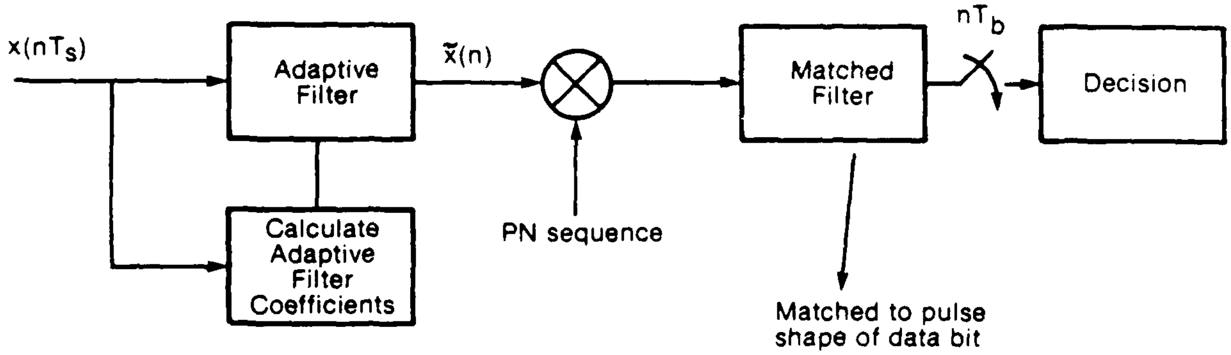


Frequency response of adaptive filter.
(e)



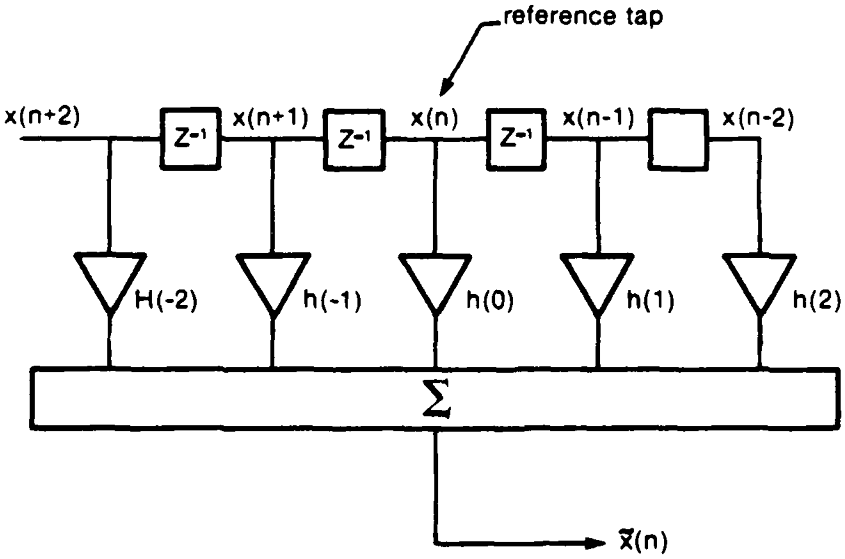
Inverse Discrete Fourier Transform (DFT) of spectrum in (e), leading to a linear phase FIR filter with deiay (M-1)/2.
(f)

FIGURE 1 (cont'd)



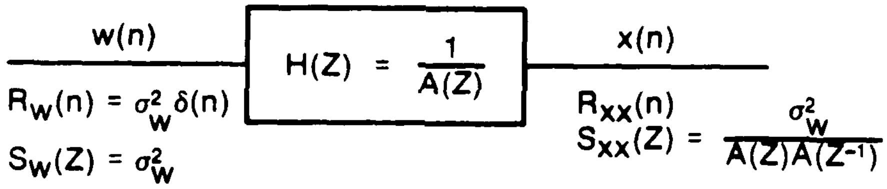
N samples from series $x(nT_s)$ = $x(n)$ are used to calculate the coefficients $h(n)$ of the adaptive filter. The same data are then processed by the filter, yielding the filtered sequence $x(n)$, consisting of the pseudo-random signal plus noise.

FIGURE 2



Five tap linear phase filter

FIGURE 3


w(n)
$R_w(n) = \sigma_w^2 \delta(n)$
$S_w(Z) = \sigma_w^2$
$H(Z) = \frac{1}{A(Z)}$
x(n)
$R_{xx}(n)$
$S_{xx}(Z) = \frac{\sigma_w^2}{A(Z)A(Z^{-1})}$

System representation of a discrete autoregressive process

FIGURE 4

3.2 Parametric Filters

The parametric filters are different from the non-parametric ones in the sense that the data samples $x(nT_s)$ are used to estimate certain parameters of a proposed model for the PSD of the time series. One such example, the one considered in this report, is the all pole model, also termed the Autoregressive (AR) or Maximum Entropy type. Here, the data is used to estimate the position of the poles. Examples of how this is actually done will be discussed later.

If a stochastic process can be modelled as AR, then, if white noise is passed through the all pole filter, the output will be the spectrum of the time series (Figure 4) $S_{xx}(z)$, with autocorrelation function $r_{xx}(n)$.

The denominator A(z) of the transfer function H(z) is

$$A(z) = 1 - a_1 z^{-1} - A_2 z^{-2} - \ldots\ldots - a_M z^{-M} \qquad 3(a)$$

Thus, the sampled signal is

$$x(n) = a_1\, x(n-1) + a_2\, x(n-2) + \ldots\ldots + a_M x(n-M) + w(n) \qquad 3(b)$$

where w(n) is a sample of white noise. Note too that equation (3b) suggests x(n) is being "predicted" by M weighted past samples, with prediction error w(n). The problem then becomes re-phrased to one involving linear prediction concepts, which AR, MA and ARMA processes are based upon [3].

The problem is to calculate the constants or weights a_i, $i = 1,2,\ldots M$ using the available data samples. Knowledge of these constants allows one indirectly to obtain an estimate of the poles from the transfer function

$$H(z) = \frac{1}{A(z)} = \frac{1}{1-a_1 z^{-1}-a_2 z^{-2} -\ldots\ldots- a_M z^{-M}} \qquad (4)$$

There are a variety of ways to calculate the coefficients a_i. First consider the theoretical case in which the data samples x(n) are random variables with known autocorrelation function $r_{xx}(n)$. Multiplying equation (3b) by x(n-k) and taking the expectation of both sides yields (assuming x(n) is a stationary, zero mean Gaussian process)

$$r_{xx}(k) = a_1 r_{xx}(k-1) + a_2 r_{xx}(k-2) + \ldots + a_M r_{xx}(k-M) + R_{wx}(k) \qquad (5)$$

Now $R_{wx}(k) = 0, (E\{w(n)x(n-k)\} = 0)$ since the noise sample at time 'n' is uncorrelated with the past output $x(n-k)$, $k > 0$, (the system we are considering is causal so that inputs at 'n' have no effect on past outputs). Thus we have a set of M equations and M unknowns a_i, $i = 1, 2, \ldots\ldots M$, i.e.,

$$\begin{array}{ccccc} a_1\ r_{xx}(0) + \ldots + a_M\ r_{xx}(1-M) & = & r_{xx}(1) \\ \vdots \qquad\qquad \vdots & & \vdots \\ a_1\ r_{xx}(M-1) + \ldots + a_M\ r_{xx}(0) & = & r_{xx}(M) \end{array} \qquad (6)$$

These are known as the Yule-Walker equations. Since $r_{xx}(n-k) = r_{xx}(k-n)$, their coefficient matrix is Toeplitz in form. In matrix notation (6) becomes

$$\underline{r}_{xx} a = \underline{r} \qquad (7)$$

where

$$\underline{r}_{xx} = \begin{bmatrix} r_{xx}(0) & r_{xx}(1) \ldots\ldots\ldots & r_{xx}(M-1) \\ r_{xx}(1) & & \cdot \\ \cdot & & \cdot \\ \cdot & & \cdot \\ \cdot & & \cdot \\ \cdot & & r_{xx}(1) \\ r_{xx}(M-1) & r_{xx}(2) & r_{xx}(0) \end{bmatrix}, \quad \underline{a} = \begin{bmatrix} a_1 \\ a_2 \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ a_M \end{bmatrix}, \quad \underline{r} = \begin{bmatrix} r_{xx}(1) \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ r_{xx}(M) \end{bmatrix}$$

If the discrete autocorrelation function is known exactly at various time lags 'k', then the exact a_i can be calculated from (7). Realistically, however, the exact autocorrelation function is not known and, therefore, must be estimated using sample realizations of the time series. A typical formula to determine the sample autocorrelation is

$$\hat{r}_{xx}(k) = \sum_{n=1}^{N-k} x(n)\ x(n-k) \qquad (8)$$

Therefore, the matrix equation (7) becomes

$$\underline{\hat{r}}_{xx} \, \underline{\hat{a}} = \underline{\hat{r}} \qquad (9)$$

where the '$\hat{}$' refers to estimate. The $\hat{a}_i$ have now become estimates for the exact a_i and, hence, indirect estimates for the poles in equation (4).

Because it is Toeplitz, the coefficient matrix $\underline{\hat{r}}_{xx}$ can be inverted efficiently compared to standard techniques such as Gaussian elimination and Cholesky LU decomposition. This efficient algorithm is known as the Levinson-Durbin algorithm [2].

Box and Jenkins have cautioned [2] that using equation (8) with the Levinson-Durbin algorithm may lead to numerical stability problems; but Makhoul [3] suggested that this is not a problem in practice.

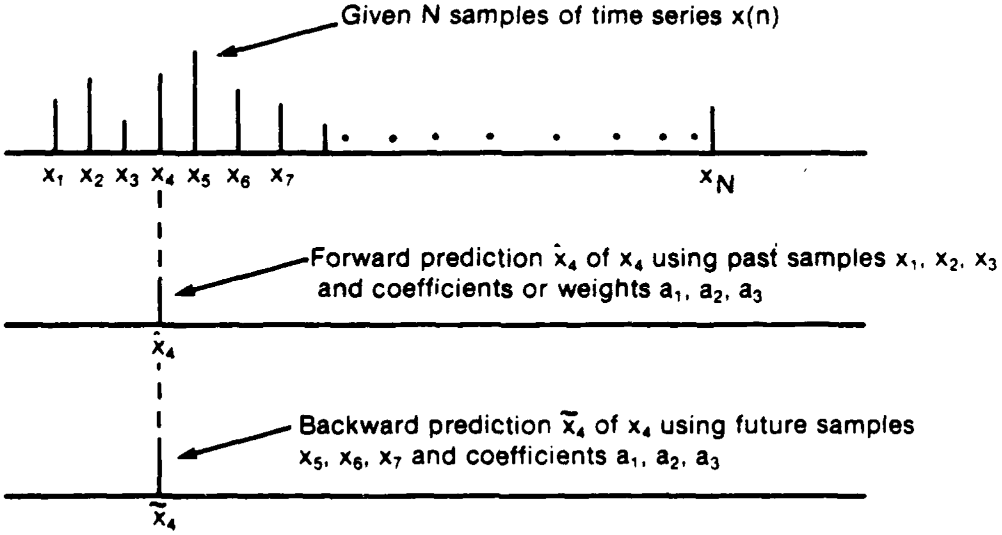
Another method of estimating the AR parameters is known as the Burg technique [4], which does not require prior estimates of the autocorrelation function of the time series. In this algorithm, it is assumed that x(n) can be estimated (or predicted) by a weighted sum of M previous samples and a weighted sum of M future samples using the same weights a_i in both directions, thus implying a stationary series (e.g., Figure 5, M = 3). The forward and backward prediction errors are minimized with respect to the M^{th} coefficient a_M and, because of the nature of the Levinson-Durbin algorithm, is used to generate the coefficients $a_{M-1}, \ldots, a_1$.

A third approach to estimating the filter weights is via the least squares method. It is also one which has been implemented in a simulation. The implementation is based on a paper by Barrodale and Erickson [5] and is similar to Burg's algorithm. The basic difference is that all coefficients a_i are determined through minimizing the sum of squares of the forward and backward residuals. The technique will now be summarized.

Given a time series $x_1, x_2, \ldots., x_N$, assume that x_t can be estimated by x_t which is a linear combination of M past values of x_t, (the forward prediction problem), i.e.,

$$x_t = \sum_{j=1}^{M} \tilde{a}_j \, x_{t-j}, \quad t = M + 1, M + 2, \ldots, N \qquad (10)$$

There is an error $\tilde{e}_t$ between x_t and $\hat{x}_t$. The parameters $\tilde{a}_j$ are determined by minimizing the residual sum of squares $\sum_{t=M+1}^{N} \tilde{e}_t^2$. Thus


Given N samples of time series x(n)
x_1 x_2 x_3 x_4 x_5 x_6 x_7
x_N
Forward prediction $\hat{x}_4$ of x_4 using past samples x_1, x_2, x_3 and coefficients or weights a_1, a_2, a_3
$\hat{x}_4$
Backward prediction $\tilde{x}_4$ of x_4 using future samples x_5, x_6, x_7 and coefficients a_1, a_2, a_3
$\tilde{x}_4$

An example of forward and backward prediction using the same weights in both directions. In this example the filter order is M = 3.

FIGURE 5

$$x_t \;=\; \sum_{j=1}^{M} \tilde{a}_j \, x_{t-j} + \tilde{e}_t, \; t = M + 1, \; \ldots, \; N \qquad (11)$$

In matrix notation (11) becomes

$$\underline{\tilde{e}} \;=\; \underline{\tilde{y}} - \tilde{X} \, \underline{\tilde{a}} \qquad (12)$$

where

$$\underline{\tilde{X}} = \begin{bmatrix} x_M & x_{M-1} \;\cdots\; x_1 \\ x_{M+1} & x_M \\ \cdot & \cdot \\ \cdot & \cdot \\ \cdot & \cdot \\ \cdot & \cdot \\ \cdot & \cdot \\ x_{N-1} & x_{N-2} \;\cdots\; x_{N-M} \end{bmatrix}, \; \underline{\tilde{a}} = \begin{bmatrix} \tilde{a}_1 \\ \tilde{a}_2 \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \tilde{a}_M \end{bmatrix}, \; \tilde{y} = \begin{bmatrix} x_{M+1} \\ x_{M+2} \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ x_N \end{bmatrix}, \; \underline{\tilde{e}} = \begin{bmatrix} \tilde{e}_{M+1} \\ \tilde{e}_{M+2} \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \tilde{e}_N \end{bmatrix}$$

A least squares solution $\tilde{a}_\star$ is defined as any $\tilde{a}$ which minimizes the residual sum of squares $S = \tilde{e}^T\tilde{e}$, where T represents the transpose. For an over determined set of equations (i.e., more equations than unknowns), a least squares solution will exist and also will be unique if the rank of $\tilde{X}$ is full (the columns of $\tilde{X}$ are linearly independent). Applying this condition to (12), S is minimized when $\tilde{X}^T\tilde{e} = 0$ (the columns of $\tilde{X}$ are orthogonal to $\tilde{e}$), and so the LS solution $\tilde{a}_\star$ must satisfy the equation

$$\tilde{X}^T \, (\tilde{y} - \tilde{X}\tilde{a}_\star) \;=\; 0 \qquad (13)$$

or $$\tilde{X}^T \, \tilde{X} \, \tilde{a}_\star \;=\; \tilde{X}^T\tilde{y} \qquad (14)$$

The LS solution is therefore characterized by the MxM system of normal equations (i.e., the solution $\tilde{a}_\star$ yields a residual vector $\tilde{e}_\star$ that is normal to the column space of $\tilde{X}$, hence the terminology).

The result in (14) has been based on the forward prediction of x_t (equation (9)) on its M past values. One can also predict in the backward direction given a set of data $x_1, x_2, \ldots, x_N$, by starting with $x_N, x_{N-1}, x_{N-2}, \ldots$, i.e.,

$$\bar{x}_{N-M} \;=\; \sum_{j=1}^{M} \bar{a}_j \, x_{N-M+j} \tag{15}$$

$$\bar{x}_1 \;=\; \sum_{j=1}^{M} \bar{a}_j \, x_{1+j}$$

and define the backward residual $e_t = x_t - \bar{x}_t$. The LS solution to the backward prediction problem is

$$\bar{X}^T \bar{X} \bar{a}_* \;=\; \bar{X}^T \bar{y} \tag{16}$$

where

$$\bar{X} = \begin{bmatrix} x_2 & x_3 & x_{M+1} \\ x_3 & x_4 & \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ x_{N-M+1} & x_{N-M+2} & x_N \end{bmatrix}, \quad \bar{a} = \begin{bmatrix} \bar{a}_1 \\ a_2 \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \bar{a}_M \end{bmatrix}, \quad \bar{y} = \begin{bmatrix} x_1 \\ \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ x_{N-M} \end{bmatrix}$$

As in Burg's algorithm, it is assumed that x_t can be estimated by a weighted sum of M previous observations and a weighted sum of M future observations (because of the stationary assumption) using the same weights a_j in both directions (see Figure 5). Thus, the LS problem can be described by the 2(N-M)xM matrix equation

$$\underline{e} \;=\; \underline{y} - \underline{Xa} \tag{17a}$$

or

$$\begin{bmatrix} \tilde{e} \\ \\ \bar{e} \end{bmatrix} = \begin{bmatrix} \tilde{y} \\ \\ \bar{y} \end{bmatrix} - \begin{bmatrix} \tilde{X} \\ \\ \bar{X} \end{bmatrix} \underline{a} \tag{17b}$$

The LS solution to (17a) satisfies the MxM system of normal equations

$$\underline{X}^T \underline{X} \, \underline{a}_* \;=\; \underline{X}^T \underline{y} \tag{18}$$

where $\underline{X}$ and $\underline{y}$ have been defined in (17b). Substituting for $\underline{X}$ and $\underline{y}$ yields the equation

$$(\tilde{X}^T\tilde{X} + \bar{X}^T\ \bar{X})\underline{a}_* \ = \ \tilde{X}^T\tilde{y} + \bar{X}^T\bar{y} \qquad (19)$$

where $\tilde{X}$, $\tilde{y}$ and $\bar{X}$, $\bar{y}$ have been defined in (12a) and (16) respectively.

The paper from which the above summary was extracted [5], contains a FORTRAN program which efficiently solves for $\underline{a}_*$, the LS solution in (19), given the sampled data sequence x_1, x_2, ..., x_N. The FORTRAN program puts no restriction on N per se (other than the fact that N must be selected so that an over determined set of equations is obtained), since the solution to (19) is carried out in a recursive manner. However, the basic thrust of any Maximum Entropy technique to solving for the coefficients a_i is to obtain accurate estimates with as few data samples as possible. Reasonable estimates can be made with N being 50 to 100 samples. Ketchum and Proakis [1] suggest that one can use fewer than 50 data samples.

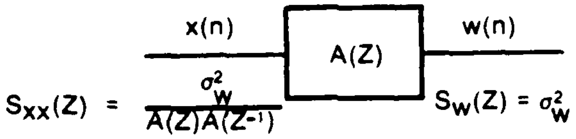
In summary, whichever AR parametric method is selected, the net result is an estimate for the coefficients a_i, i=1, 2, ..., M in equation (4). However, what has not been addressed is the problem of what value of filter order M to choose. Suffice it to say that there is a technique to choose M optimally for the methods discussed above [3]. It has been found though, from [1] that the filter order does not affect performance very much, i.e., M=4 appears to work just as well as a filter order of 15 for the case of a dense cluster of stable tone interferers in the communication band. For this reason, preliminary work herein has arbitrarily selected a fixed filter order of 4 simply to explore the kind of results that would be obtained.

Referring back to Figure 4, we can see that if $\sigma_w^2/A(z)A(z^{-1})$ adequately represents the PSD of x(n), then A(z) can be interpreted as a whitening filter once the a_i are determined (see Figure 6) where $A(z) = 1-a_1\ z^{-1} - a_2z^{-2} - a_3z^{-3} - a_4z^{-4}$ for the case of M=4, i.e., the impulse response is $h(n) = \delta(n) - a_1\ \delta(n-1) - a_2\ \delta(n-2) - a_3\ \delta(n-3) - a_4\ \delta(n-4)$. It is an FIR filter as shown in Figure 7 which does not have linear phase since the impulse response is not symmetric. Note too that there is no delay through this filter.

4.0 INPUT SIGNAL CHARACTERISTICS

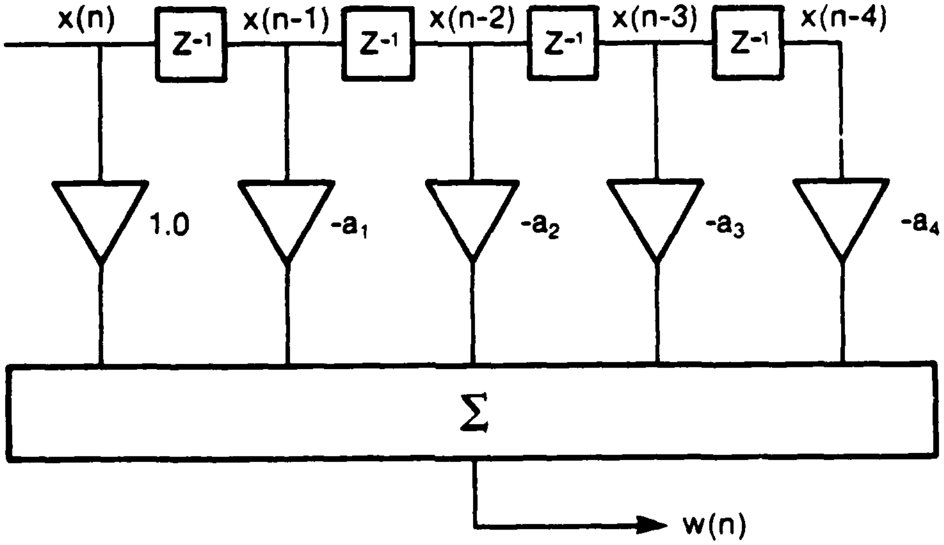
At this stage, a few comments will be made about the input signal x(n). It basically consists of three components:

(i) the direct sequence signal s(n)
(ii) Gaussian noise n(n)
(iii) the interference signal i(n).



System representation of a whitening filter for an AR process

FIGURE 6



Five tap transversal filter of a whitening filter

FIGURE 7

The parametric method (and also the non-parametric method as well) is based on the premise that the autocorrelation function of x(n) is dominated by the autocorrelation function of the interference if:

(i) the signal x(n) is sampled at the chip rate and
(ii) we confine ourself to within the PN sequence period.

These two points should be elucidated.

First assume that the PN sequence is infinite in extent which implies it has an autocorrelation function as shown in Figure 8 where T_c is the chip duration. If this autocorrelation function is sampled at the chip rate, the following sampled autocorrelation function (ACF) in Figure (9) is obtained. This ACF has a flat power spectral density. (Realistically, $R_{ss}(n)$ will not be as shown in Figure 9, but will have some non-zero autocorrelation values at nT_c, n = 1, 2, ... etc., leading to a non-flat spectrum and hence some code cancellation or distortion in the adaptive filtering process.)

Second, the additive noise is white Gaussian and has an ACF consisting of the Dirac impulse function whose spectrum is flat.

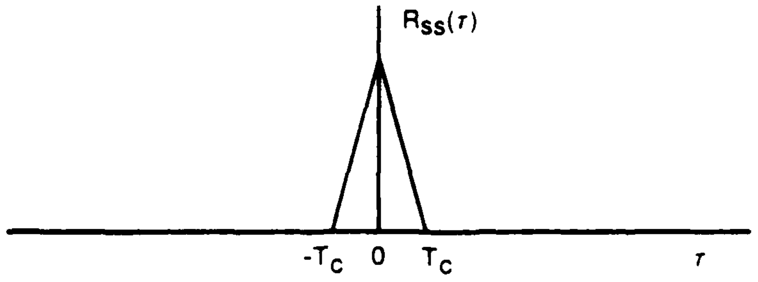
Third, the interference will have an ACF extending beyond $\pm T_c$ and therefore will have a non-flat PSD. Figure 10 shows the sum of the three ACF's and their resultant PSD's for the case of a single stable tone. Ideally, the notch filter in Figure 7 should attenuate the tone in Figure 10(b) quite significantly. This can be seen intuitively.

For example, referring to Figure 7

$$w(n) = x(n) - a_1\, x(n-1) - a_2\, x(n-2) - a_3\, x(n-3) - a_4\, x(n-4) \tag{20a}$$

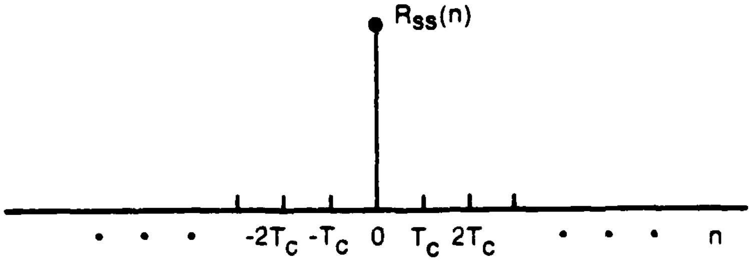
$$\begin{aligned} &= [s(n) - a_1\, s(n-1) - a_2\, s(n-2) - a_3\, s(n-3) - a_4\, s(n-4)] + \\ &\quad + [n(n) - a_1\, n(n-1) - a_2\, n(n-2) - a_3\, n(n-3) - a_4\, n(n-4)] + \\ &\quad + [i(n) - a_1\, i(n-1) - a_2\, i(n-2) - a_3\, i(n-3) - a_4\, i(n-4)] \end{aligned} \tag{20b}$$

It can be seen from the third term in square brackets in (20b) that if i(n) is a reasonably good linear combination of its past four values, then the interference will be cancelled to a large extent. Similarly, if s(n) and n(n) cannot be well represented by their past values then little cancellation will occur.



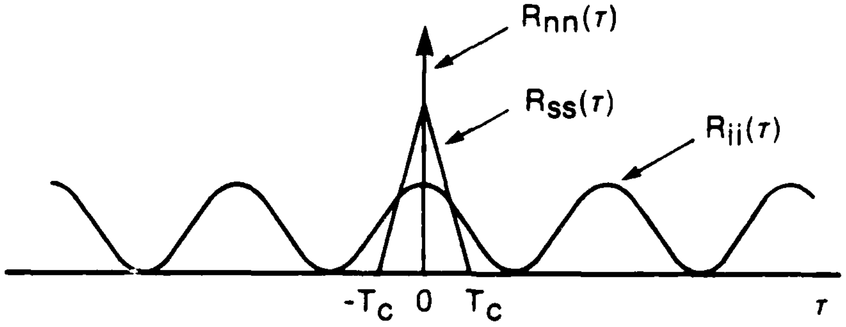
Autocorrelation function of an infinite random sequence.

FIGURE 8



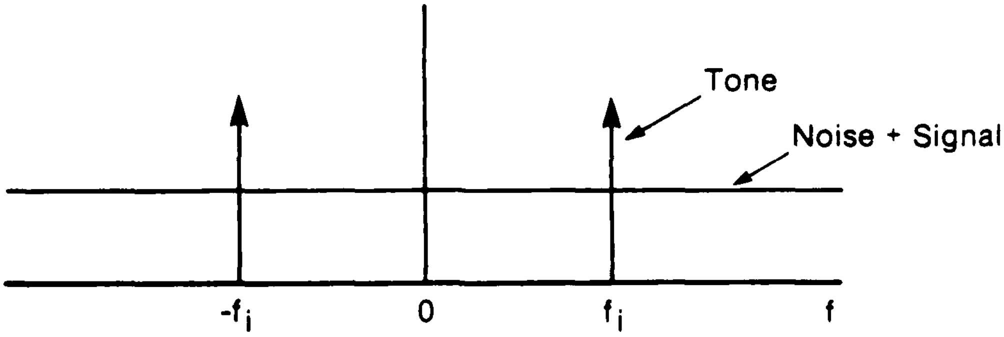
Sampled autocorrelation function of Figure 8, sampled at the chip rate.

FIGURE 9



Autocorrelation function of white noise, random PN sequence and sinusoidal interference

FIGURE 10 (a)



Fourier transform of the autocorrelation function in Figure 10 A)

FIGURE 10 (b)

5.0 SIMULATION RESULTS

5.1 Introduction

In the paper by Ketchum and Proakis [1] the performance of block non-parametric and parametric techniques were compared.

(i) Non-parametric

- the Welch FFT method was used to estimate the PSD from a sampled time series
- the resultant PSD was sampled and its reciprocal calculated
- the impulse response of the filter was determined thus resulting in tap weights for a linear phase FIR filter

(ii) Parametric (three techniques were compared)

- the autocorrelation method
- the Burg algorithm
- a Least Squares algorithm

In the non-parametric case many more samples (roughly 900) were required to obtain a good, sharp estimate of the interfering signal, whereas reasonably good results were obtained with 50 samples (the lowest number examined) using any of the parametric methods.

Two types of interfering signal were examined:

(i) multiple stable tones (empirical results for 10 tones occupying 20% of the band and theoretical results for 100 tones)
(ii) narrowband colored noise obtained by passing white Gaussian noise through a Butterworth filter with a sharp cutoff.

The paper [1] did not address the non-stationary interferer, but did make passing reference to the fact that the effectiveness of any of the block techniques examined would depend on the block size.

To handle this other type or interferer, other adaptive schemes could also be examined whose filter coefficients are continually being updated. Examples include:

(i) Widrow-Hoff LMS algorithm [9]
(ii) Fast Kalman [10]
(iii) Recursive Least Squares techniques [11]

There has been some work on the application of (i) to the interference rejection problem [8]. However, it suffers to some extent on convergence rate, i.e., the faster the tone sweeps the more problems it encounters. This suggests examining (ii) which has a faster convergence rate than (i). No references seem to exist on its application to interference rejection. There appears to be potential in (iii). Thus, based on this preliminary assessment of the application of (i) to (iii) to the interference rejection problem (particularly for swept tone or FM type interfering signals), there appears to be some scope for exploratory work.

5.2 Experimental Set-Up

Two adaptive schemes have been implemented on the computer (PDP 11/60 and FPS AP120B). The scheme selected is block parametric processing: the Least Squares algorithm and the autocorrelation method, in particular equations (19) and (7).

The simulation is based on [1]. The following assumptions are made:

(i) the channel has infinite bandwidth
(ii) the energy/chip is unity and the amplitude is ±1
(iii) the processing gain ranges from 20 to 60
(iv) the signal-to-interference ratio per chip is -20 dB
(v) the incoming signal at the receiver is sampled at the chip rate.

Let us examine the implications of (i) to (v):

- item (i) theoretically implies that no chip distortion results (square chips are assumed). A random sequence of ± 1's has an ACF which is triangular and a PSD which is a $(\text{sinc})^2$ function (Figure 11).

- (ii) implies that $T_c = 1$. This then implies that $R_c = 1$, i.e., the chip rate.

- (iii) implies the energy/bit is equal to the processing gain, since the number of chips/bit is equal to the processing gain.

- (iv) suggests that for a single tone interferer of amplitude B, SIR = $-20 = 10 \log (\text{signal power/chip})/B^2/2 = 10 \log (E_c/T_c)/(B^2/2)$. This implies the sinusoidal amplitude is 14.14.

- (v) implies we do not have to worry about the Nyquist criterion, which applies to finite bandwidth systems and is used in designing communications systems to minimize inter-symbol interference. In the simulation model no inter-symbol interference occurs and there is no path delay occurring over the channel. Finally, sampling at the chip rate, R_c, implies a Nyquist bandwidth of $R_c/2$. Since there is no filtering anywhere (except for the adaptive filter which comes later), any interfering signals outside the range $R_c/2$ will be aliased back into $-R_c/2 < f < R_c/2$. Therefore, all interfering signals are kept in the $0 < f < R_c/2$ range. Also, since $T_c = 1$ in the simulation $R_c/2 = 1/2$.

In one of the simulation examples 10 tones are equally spaced. The frequency range is 0.0 to 0.1 Hz (Figure 12), implying the tones cover 20% of the band. The equation for the interference is

$$i(t) = 14.14 \sum_{k=1}^{10} \cos(2\pi f_k t + \theta_k) \tag{21}$$

where θ_k is a random phase between 0 and 2π. The sampled interference is

$$i(nT_c) = 14.14 \sum_{k=1}^{10} \cos(2\pi f_k nT_c + \theta_k) \tag{22}$$

At this point a few words should be said about the signal to noise ratio per bit, E_b/N_o.

The processing gain, say, is PG, and since the energy per chip is 1 and there are PG chips/bit, $E_b = PG$. The single-sided noise power spectral density N_o (Figure 13) is $2\sigma_n^2$, where σ_n^2 is the noise power. Therefore, $E_b/N_o = PG/2\sigma_n^2$. Thus for a given E_b/N_o in dB, the noise variance σ_n^2 can be determined. For BPSK and no interference the error rate is

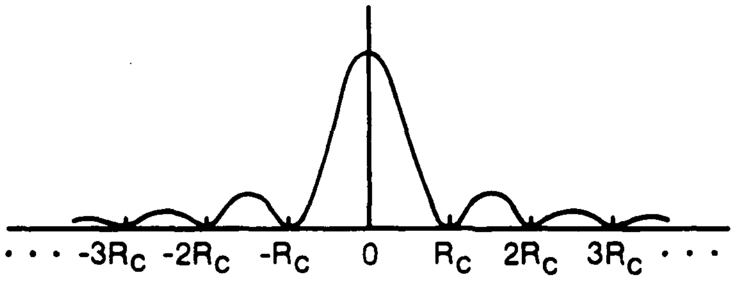
$$P_b = \frac{1}{2} \operatorname{erfc}\sqrt{\frac{E_b}{N_o}} = \frac{1}{2} \operatorname{erfc}\sqrt{\frac{PG}{2\sigma_n^2}} \tag{23}$$

We now have all of the variables to generate a time series of N points, i.e.,

$$x(nT_c) = \pm 1 + n(nT_c) + 14.14 \sum_{k=1}^{10} \cos(2\pi f_k nT_c + \theta_k) \tag{24}$$

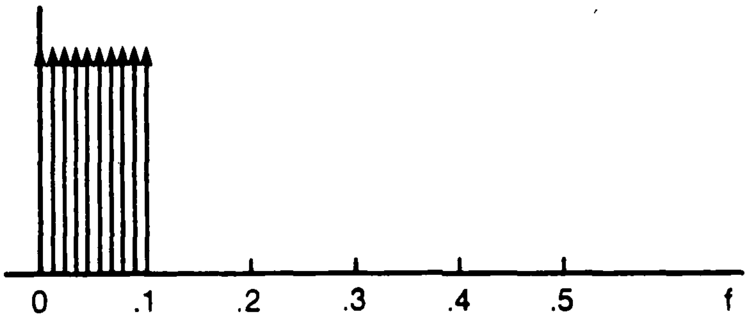
where $n(nT_c)$ is a sample of noise from a Gaussian distribution with variance σ_n^2 determined from above, i.e., from $E_b/N_o = PG/2\sigma_n^2$.

A block diagram of the simulation set-up is shown in Figure 14. The simulation generates 200,000 chips which implies 10,000 bits (PG = 20) for E_b/N_o's ranging from -10 dB to +15 or 20 dB. Before preliminary results are presented, some spectral plots based on the autocorrelation method will be presented.



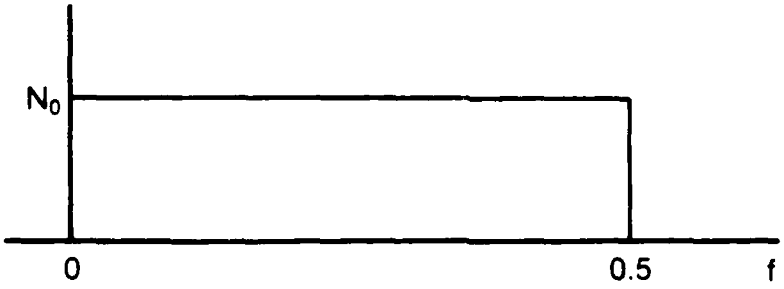
Power spectral density function of infinite random sequence.

FIGURE 11



Single-sided spectrum showing 10 interfering stable tones occupying 20% of the band. The tones are separated in frequency by .0111 Hz and are of equal amplitude.

FIGURE 12



One-sided noise PSD.

FIGURE 13

5.3 Test Results

The upper curve in Figure 15(a) is a spectral estimate of a single tone interferer located at 0.25 Hz and is based on a 4-pole model, i.e.,

$$H(\omega) = \frac{1}{1 - a_1 e^{-j\omega} - a_2 e^{-j2\omega} - a_3 e^{-j3\omega} - a_4 e^{-j4\omega}} \tag{25}$$

where the numerator has been equated to 1 for now. The coefficients a_i were estimated using the autocorrelation technique (equation (9)) with the Levinson-Durbin algorithm. The correlation matrix was generated using data form equation (24) for a single tone and a noise variance $\sigma_n{}^2$ of 0.01, which corresponds to E_b/N_o = 30 dB when the processing gain is 20.

The lower curve in Figure 15(a) is the spectral estimate obtained by cascading the model H(ω) with its matched filter H*(ω) as shown in Figure 16. The upper curve in Figure 15(b) shows the frequency response of the adaptive filter without its matched filter, i.e.,

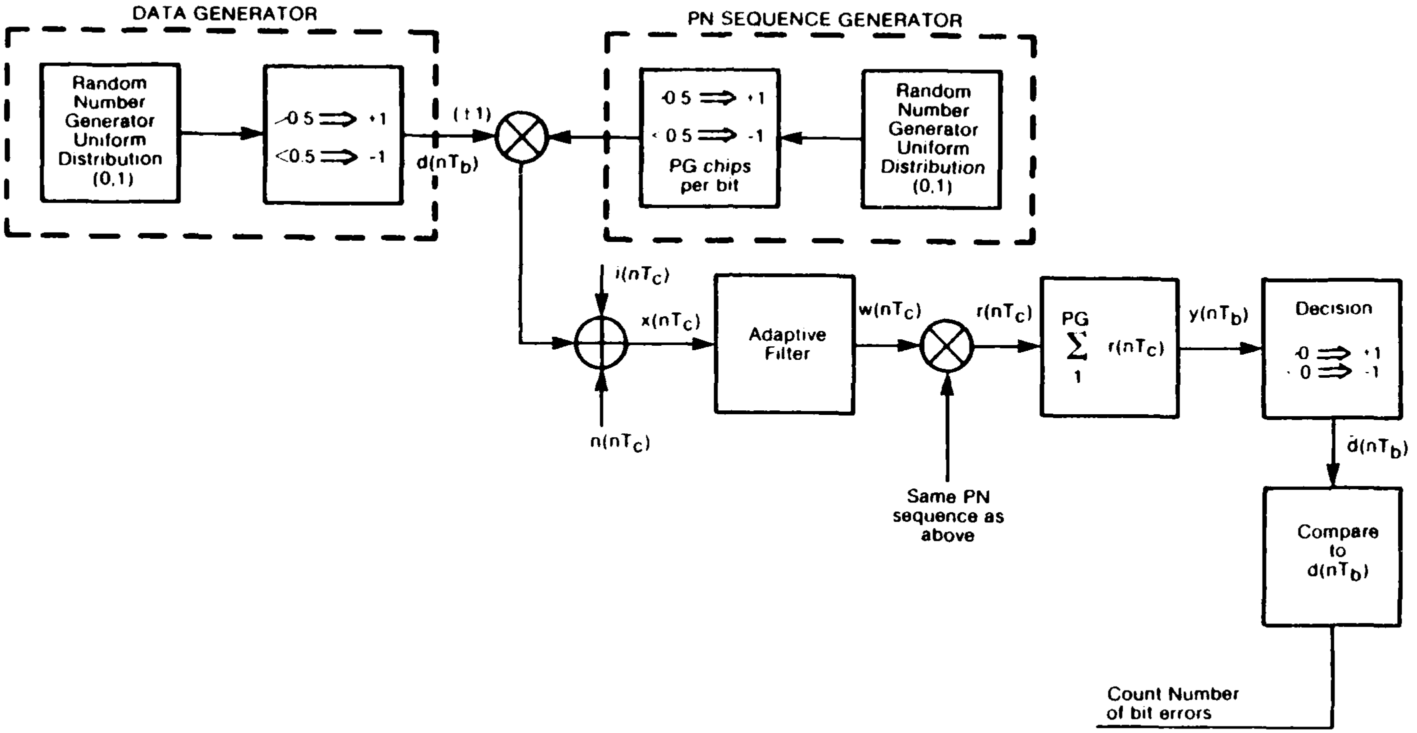
$$F(\omega) = \frac{1}{H(\omega)} = 1 - a_1 e^{-j\omega} - a_2 e^{-j2\omega} - a_3 e^{-j3\omega} - a_4 e^{-j4\omega} \tag{26}$$

The inverse transform of equation (26) is the hypothetical finite impulse response

$$f(n) = 1 - a_1\,\delta(n-1) - a_2\,\delta(n-2) - a_3\,\delta(n-3) - a_4\,\delta(n-4) \tag{27}$$

in Figure 17. One can design an adaptive filter using these weights, but better performance will be obtained by cascading f(n) with its matched response f*(4-n) = f(4-n) [1]; the result will lead to a linear phase filter with weights calculated from the convolution of f(n) and f(4-n) shown in Figure 18. The notch filter for the single sinusoid case is the bottom curve in Figure 15(b). Note too that the combined adaptive filter with its matched filter introduces a delay in the output, which must be kept in mind when correlating the filtered output with the PN sequence.

Figure 19 shows plots of AR spectra for the case of ten equi-amplitude tones ranging in frequency from 0.2 Hz to 0.3 Hz, thus covering 20% of the band. The frequency spacing between sinusoids is approximately *0.0111 Hz.* The spectrum plotted in Figure 20(a) was for the time series in Figure 19, but modelled as AR-10: 50 samples were used to estimate the pole positions. Figure 20(b) is AR-40 using 200 time samples. Figure 21 shows two spectral samples based on an AR-4 model and 2 different blocks of 50 data samples used to estimate the filter coefficients.



Block Diagram of the simulation set-up

FIGURE 14

Figure 22 compares the theoretical E_b/N_o vs P_b (bit error rate) for the AWGN channel with the simulation program developed on the PDP11/60 and FPS AP-120. One sample run produced the results indicated by 'o'. A processing gain of 20 was used and 200,000 data samples were generated implying 10,000 data bits (there are 20 samples/bit since the sampling rate is at the chip rate). The number of bit errors were determined at the various E_b/N_o's shown, by taking the ratio of the number of bit errors to the number of bits transmitted. No interference was added to the channel. It should be noted too that the results in Figure 22 and subsequent figures were obtained with only one run of the simulation program, not several as it theoretically should be for Monte Carlo simulations. The results, however, differ negligibly from run to run for the examples considered herein.

Figure 23 shows the bit error rate performance when 10 equi-amplitude tones ar^ present, spanning 20% of the band. The frequency range is from 0.0 to 0.1 Hz, with a frequency spacing of 0.0111 Hz between tones. There are four curves presented and were generated under the following conditions:

(i) processing gain of 20
(ii) number of samples/block for estimating the filter coefficients was 100 using the autocorrelation technique and Levinson-Durbin algorithm
(iii) the filter order was 4.

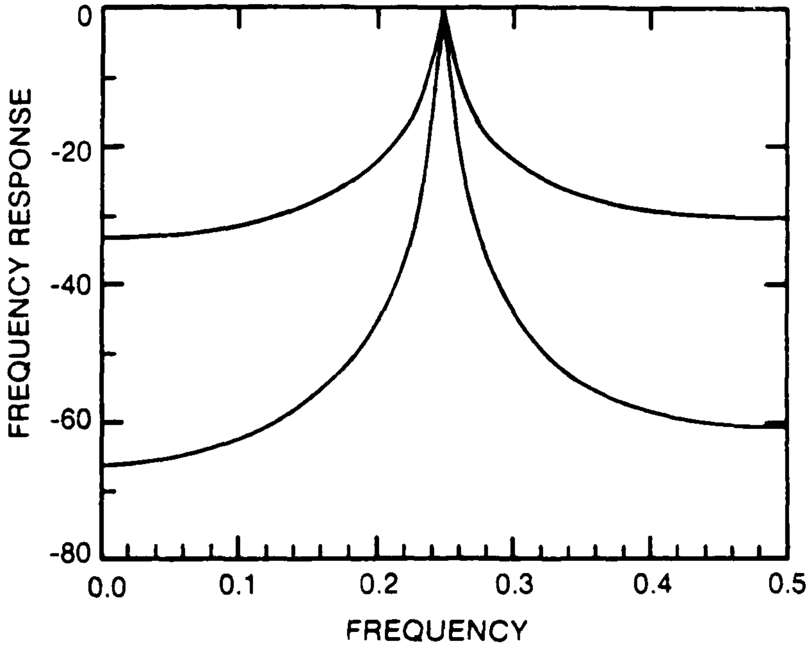
One noteworthy point about the results in Figure 23 is the improvement in performance when the adaptive filter with its matched filter are used to attenuate the interference. Figure 24 shows frequency response samples of the adaptive filter with (lower curve) and without (upper curve) its matched filter calculated for the following conditions:

(i) PG=20
(ii) number of points/block = 100
(iii) AR-4 model
(iv) Autocorrelation method
(v) E_b/N_o = 0 dB which implies $\sigma_n^2 = 10.24$.

The effect of a stable tone and a strong swept tone on bit error rate is shown in Figure 25. The results confirm that the batch processing technique starts to falter as the tone sweeps faster. 100 data points per block were used to generate the filter coefficients. There is a slight improvement when 20 data points per block are used as shown in Figure 26.

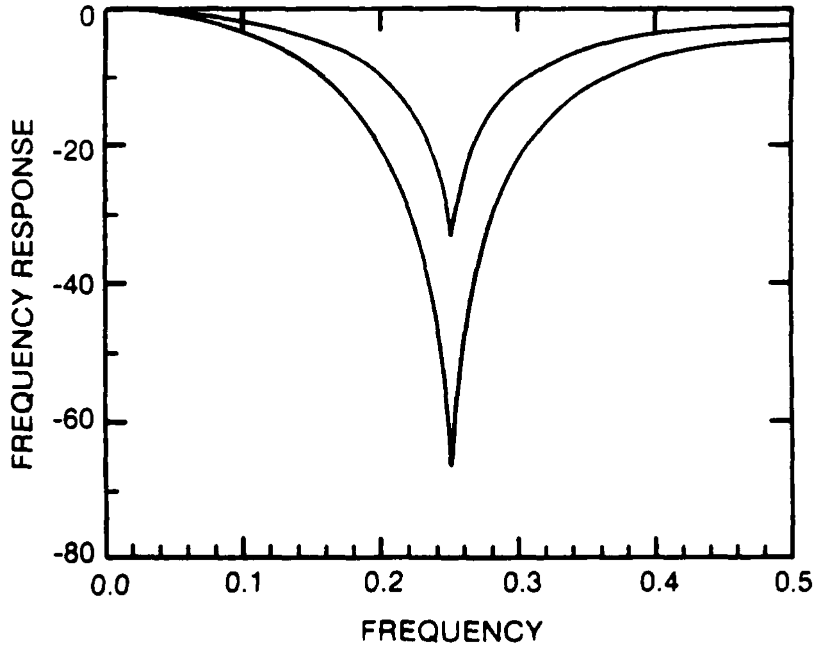
6.0 SUMMARY

A computer simulation of a BPSK DS/SS system has been developed to investigate the effects of adaptively filtering stable and swept tone interferers in the communication band.



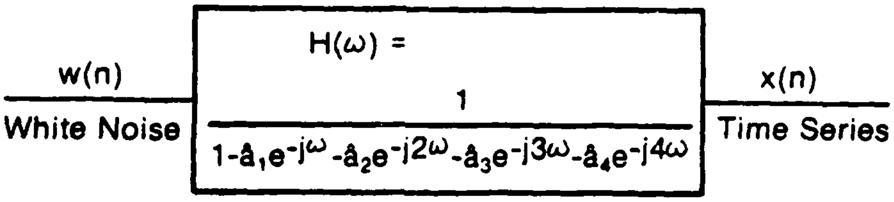
Spectral estimate of a single tone located at .25 Hz, using the autocorrelation method.

FIGURE 15(a)



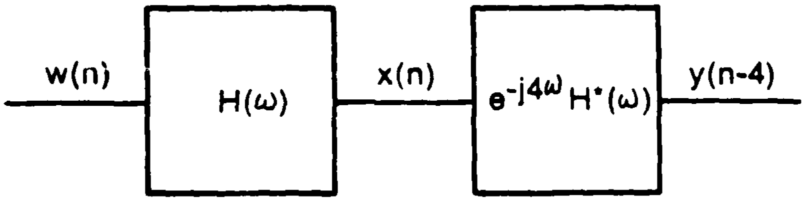
Notch filter based on spectral estimate in Figure 15(a).

FIGURE 15(b)


w(n)
White Noise
$H(\omega) = \dfrac{1}{1-\hat{a}_1 e^{-j\omega}-\hat{a}_2 e^{-j2\omega}-\hat{a}_3 e^{-j3\omega}-\hat{a}_4 e^{-j4\omega}}$
x(n)
Time Series

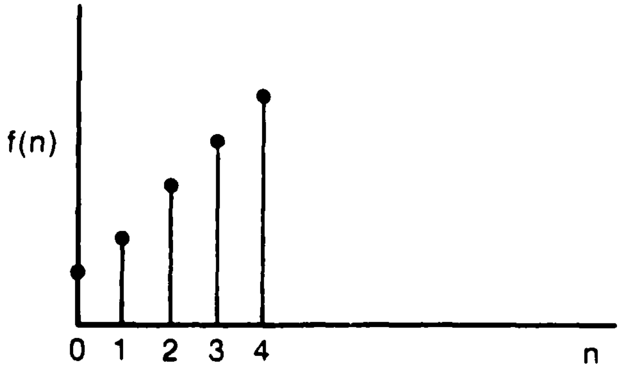
Block diagram of 4-pole spectral estimator,
without the matched filter.

FIGURE 16(a)



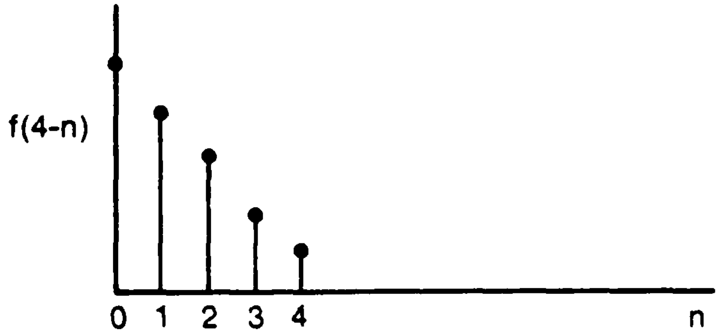
Block diagram of 4-pole spectral estimator
with its realizable matched filter.

FIGURE 16(b)



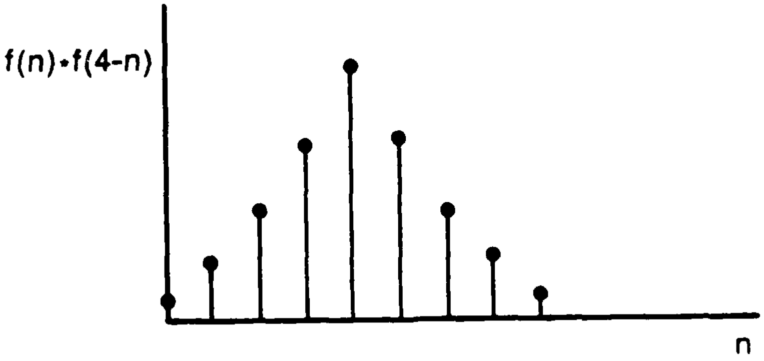
Hypothetical impulse response of a 4-zero filter.

FIGURE 17



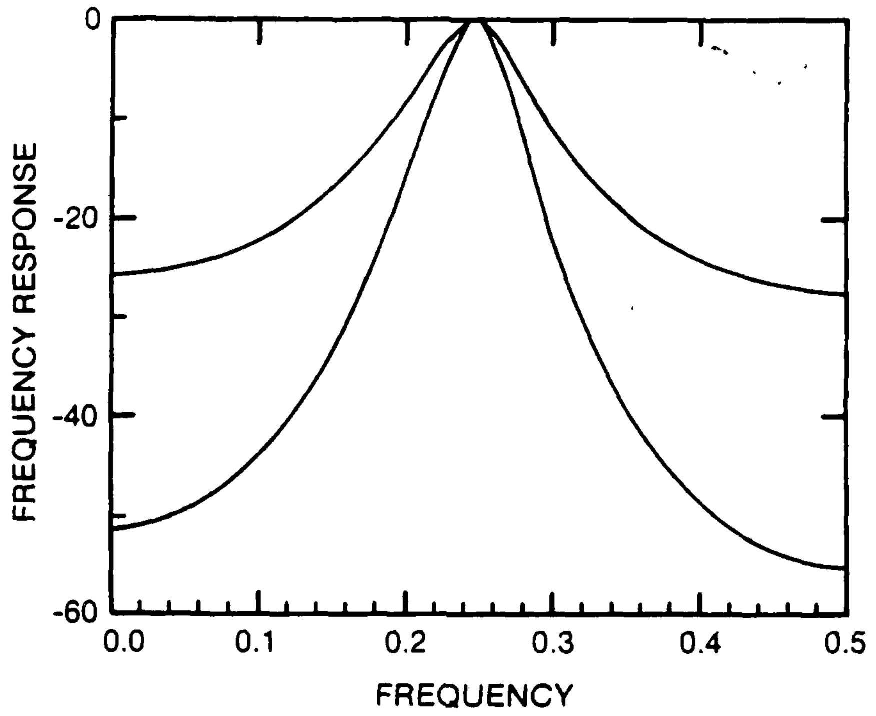
Impulse response of the realizable matched filter of Figure 17.

FIGURE 18(a)



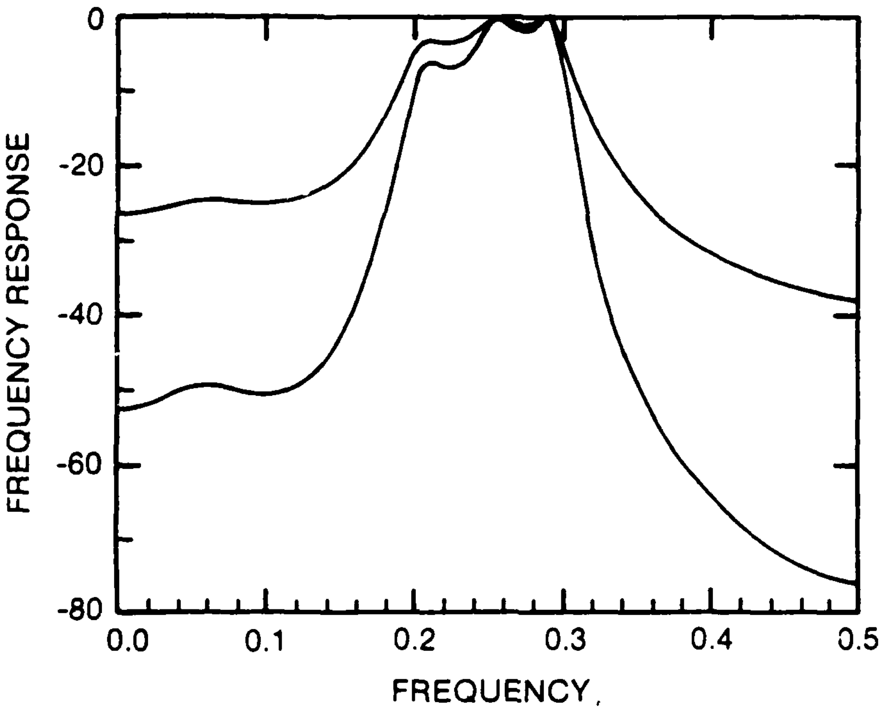
Impulse response of the adaptive filter with its matched filter obtained through convolution.

FIGURE 18(b)



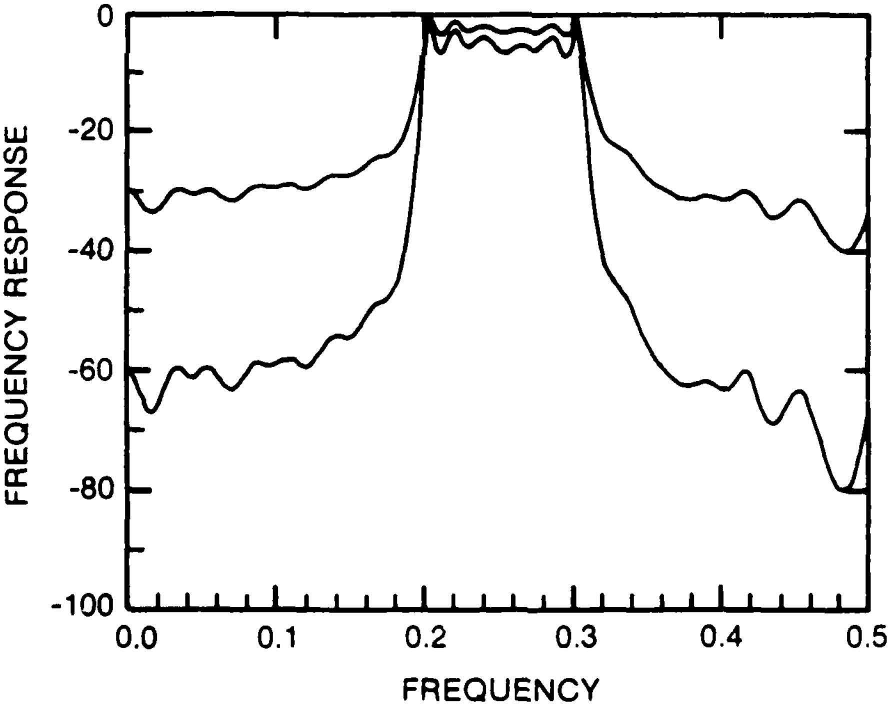
AR spectrum of a time series consisting of a pseudo-random sequence, white Gaussian noise and 10 equi-amplitude tone interferers spanning 20% of the band (0.2 to 0.3 Hz). The tones are spaced 0.0111 Hz, the noise variance σ_n^2 is 0.01 and the processing gain is 20. The model for the time series is AR-4. The pole estimates were obtained via the autocorrelation method and Levinson-Durbin algorithm, using 50 samples from the time series.

FIGURE 19



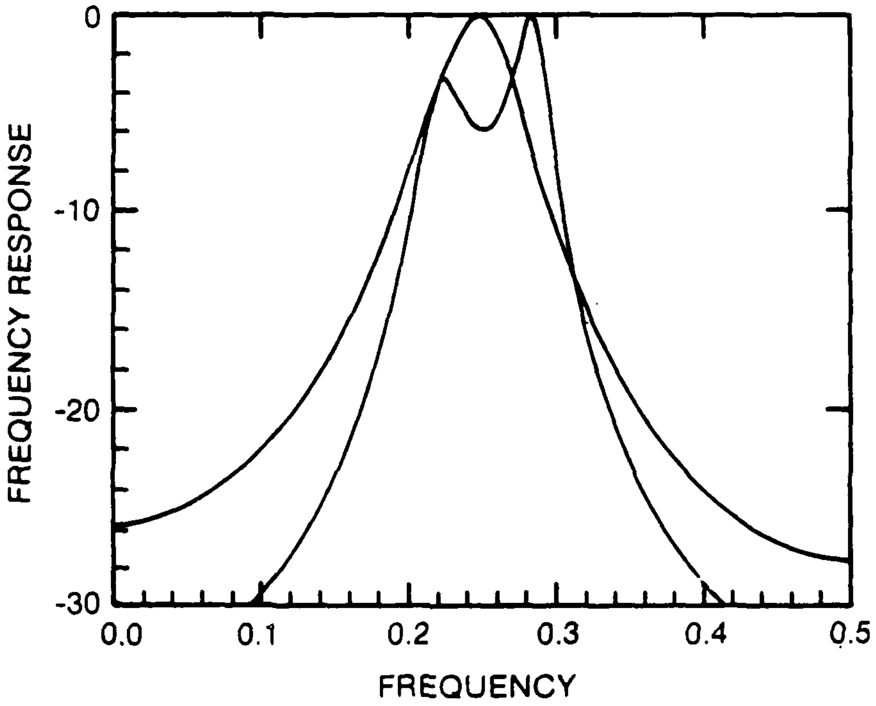
Similar to Figure 19 except the model is AR-10.

FIGURE 20 (a)



AR-40 model using 200 time samples instead of 50 as in Figures 19 and 20(a).

FIGURE 20(b)

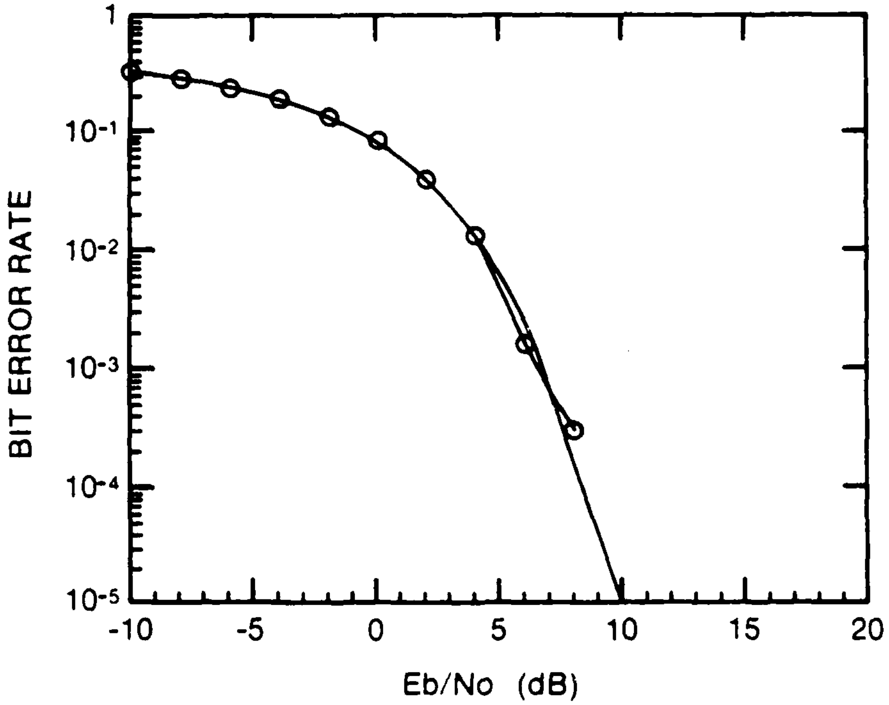


This shows the variability in the spectra from data block to data block. The conditions used to produce these spectra are the same as those in producing the upper curve in Figure 19.

FIGURE 21

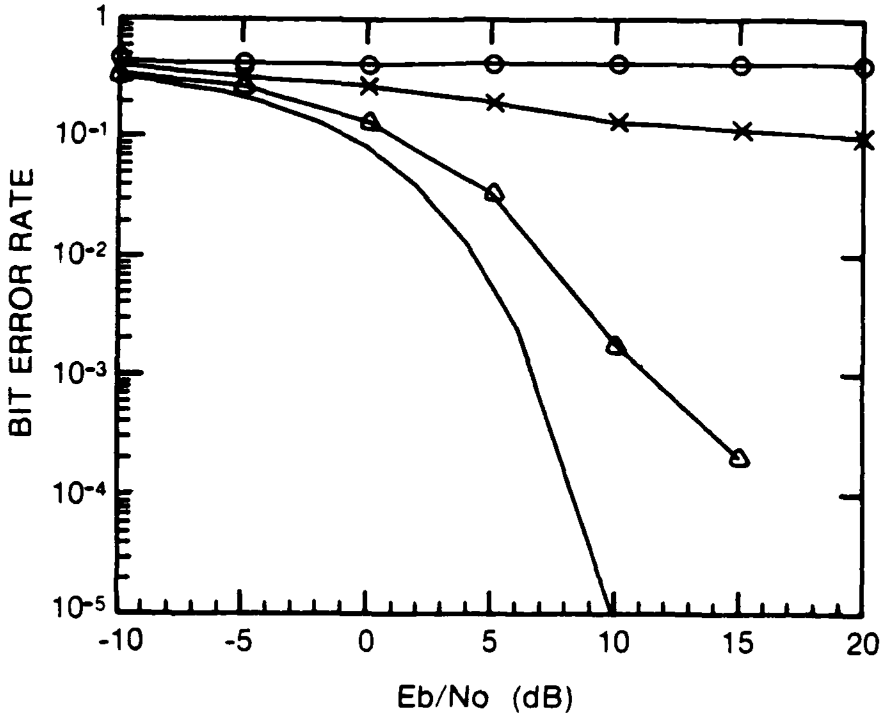
The report focusses on the block processing type of adaptive filter. It shows that this kind of filter is effective when the tones are stable, but starts to degrade in performance as the tone sweeps quickly across the band. This suggests that fewer samples per block or other types of filters should be examined, such as the

(i) Widrow-Hoff LMS type
(ii) Fast Kalman type
(iii) Recursive least squares filters.



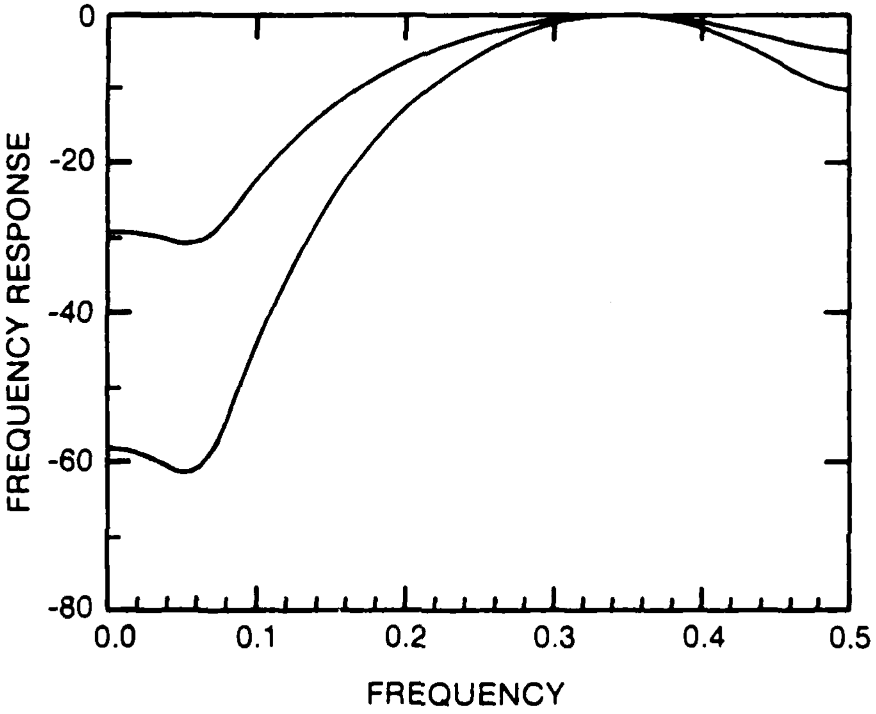
A comparison of one run of the simulation model ('o') to the theoretical bit error rate for the AWGN channel, no interference present. The processing gain was 20.

FIGURE 22



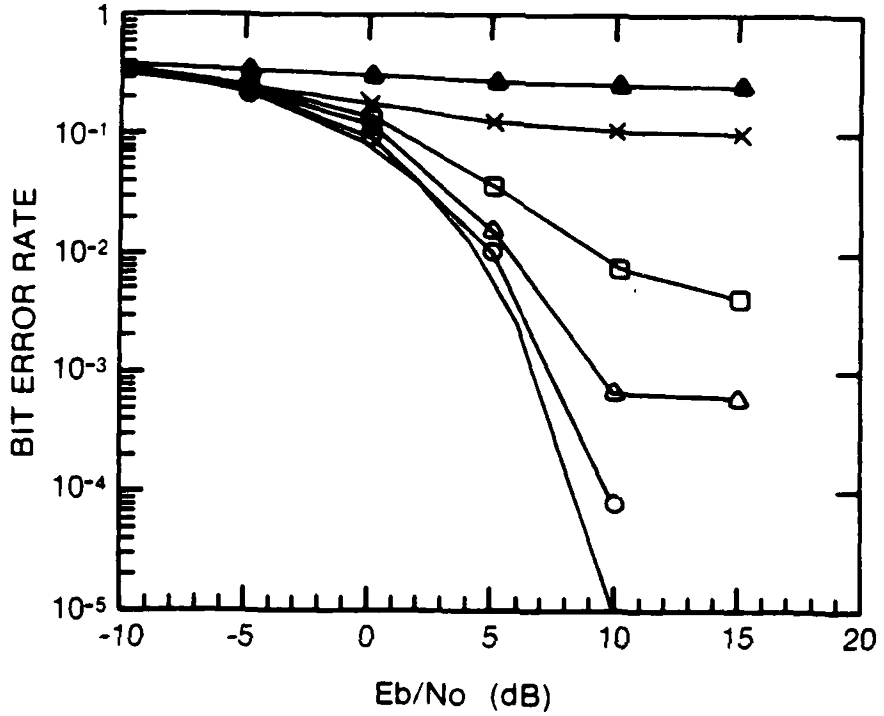
Bit error rate for three cases: (o) -- interference and no adaptive filter: (x) -- interference and adaptive filter without its matched filter: (Δ) -- interference and adaptive filter with its matched filter. Note that th interference consisted of 10 tones (0 to 0.1 Hz) and the processing gain was 20.

FIGURE 23



The upper and lower curves refer to single realizations of the adaptive filter without and with the matched filter.

FIGURE 24



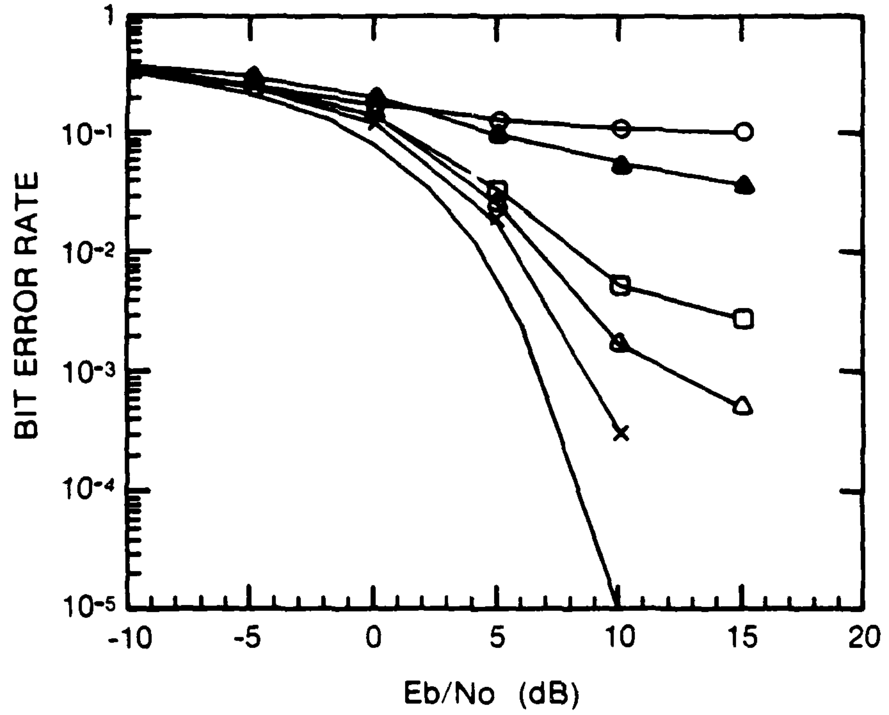
The effect on bit error rate of a non-stable interferer is exhibited here.

-(x) stable tone at 0.0 Hz, no adaptive filter

-(o) stable tone at 0.0 Hz, adaptive filter and its matched filter included.

-(Δ),(□),(▲) tone sweeping back and forth between 0.0 and 0.5 Hz at 0.0001, 0.001, and 0.01 Hz/sec., respectively; adaptive filter and matched filter included.

FIGURE 25



The conditions in this figure are similar to those in Figure 25, the only difference being that 20 data points/block were used to generate the filter coefficients.

-(x) stable tone with adaptive filter
-(o) no adaptive filter
-(Δ),(□),(▲) tone sweeping back and forth between 0.0 and 0.5 Hz at 0.0001, 0.001, and 0.01 Hz/sec., respectively; adaptive filter and matched filter included.

FIGURE 26

7.0 REFERENCES

[1] J.W. Ketchum, J.G. Proakis, "Adaptive Algorithms for Estimating and Suppressing Narrow-Band Interference in PN Spread-Spectrum Systems", IEEE Transactions on Communications, vol. COMM-30, pp. 913-924, May 1982.

[2] G.E. Box, G.H. Jenkins, "Time Series Analysis: Forecasting and Control", Holden-Day, San Francisco, 1970.

[3] J. Makhoul, "Linear Prediction: A Tutorial Review", Proceedings of the IEEE, vol. 63, pp. 561-580, April 1975.

[4] J.P. Burg, "Maximum Entropy Spectral Analyses", in Proc. 37th Meeting of the Soc. of Exploration Geophysicists, 1967; also reprinted in "Modern Spectrum Analysis", D.G. Childers, Ed. New York: IEEE Press, 1978, pp. 34-41.

[5] I. Barrodale, R.E. Erickson, "Algorithms for Least-Squares Linear Prediction and Maximum Entropy Spectral Analysis-Part 1: Theory, Part 2: FORTRAN Program", Geophysics, vol. 45, pp. 420-432, March, 1980.

[6] "Programs for Digital Signal Processing", New York: IEEE Press, 1979.

[7] R.L. Moses, J.A. Cadzow, A.A. Beex, "A Recursive Procedure for ARMA Modelling", IEEE Transactions on Acoustics, Speech and Signal Processing, vol. ASSP-33, pp. 1188-1196, October 1985.

[8] G.J. Saulnier, P.K. Das, L.B. Milstein, "An Adaptive Digital Suppression Filter for Direct-Sequence Spread Spectrum Communications", IEEE Journal on Selected Areas in Communications, Vol. SAC-3, No. 5, pp. 676-686, September 1985.

[9] B. Widrow et al., "Adaptive Noise Cancelling: Principles and Applications", Proceedings of the IEEE, vol. 63, pp. 1692-1719, Dec 1975.

[10] D.D. Falconer, L. Ljung, "Application of Fast Kalman Estimation to Adaptive Equalization", IEEE Transactions on Communications, vol. COM-26, pp. 1439-1446, October 1978.

[11] D.T.L. Lee, M. Morf, B. Friedlander, "Recursive Least Squares Ladder Estimation Algorithms", IEEE Transactions on Circuits and Systems, vol. CAS-28, pp. 467-481, June 1981.

UNCLASSIFIED

SECURITY CLASSIFICATION OF FORM
(highest classification of Title, Abstract, Keywords)

DOCUMENT CONTROL DATA

(Security classification of title, body of abstract and indexing annotation must be entered when the overall document is classified)

1. ORIGINATOR (the name and address of the organization preparing the document. Organizations for whom the document was prepared, e.g. Establishment sponsoring a contractor's report, or tasking agency, are entered in section 8.)

DEPARTMENT OF NATIONAL DEFENCE
DEFENCE RESEARCH ESTABLISHMENT OTTAWA
SHIRLEY BAY, OTTAWA, ONTARIO K1A 0Z4 CANADA

2. SECURITY CLASSIFICATION (overall security classification of the document. including special warning terms if applicable)

UNCLASSIFIED

3. TITLE (the complete document title as indicated on the title page. Its classification should be indicated by the appropriate abbreviation (S,C,R or U) in parentheses after the title.)

"APPLICATION OF AUTOREGRESSIVE METHODS TO ADAPTIVE EXCISION OF INTERFERING TONES FROM DIRECT SEQUENCE SPREAD SPECTRUM COMMUNICATIONS SYSTEMS" (U)

4. AUTHORS (Last name, first name, middle initial. If military, show rank, e.g. Doe, Maj. John E.)

KOZMINCHUK, BRIAN W.

5. DATE OF PUBLICATION (month and year of publication of document)

MARCH 1988

6a. NO. OF PAGES (total containing information. Include Annexes, Appendices, etc.)

38

6b. NO. OF REFS (total cited in document)

11

6. DESCRIPTIVE NOTES (the category of the document, e.g. technical report, technical note or memorandum. If appropriate, enter the type of report, e.g. interim, progress, summary, annual or final. Give the inclusive dates when a specific reporting period is covered.)

DREO TECHNICAL NOTE

8. SPONSORING ACTIVITY (the name of the department project office or laboratory sponsoring the research and development. Include the address.)

9a. PROJECT OR GRANT NO. (if appropriate, the applicable research and development project or grant number under which the document was written. Please specify whether project or grant)

041LK11

9b. CONTRACT NO. (if appropriate, the applicable number under which the document was written)

10a. ORIGINATOR'S DOCUMENT NUMBER (the official document number by which the document is identified by the originating activity. This number must be unique to this document.)

10b. OTHER DOCUMENT NOS. (Any other numbers which may be assigned this document either by the originator or by the sponsor)

11. DOCUMENT AVAILABILITY (any limitations on further dissemination of the document, other than those imposed by security classification)

(X) Unlimited distribution
( ) Distribution limited to defence departments and defence contractors; further distribution only as approved
( ) Distribution limited to defence departments and Canadian defence contractors; further distribution only as approved
( ) Distribution limited to government departments and agencies; further distribution only as approved
( ) Distribution limited to defence departments; further distribution only as approved
( ) Other (please specify):

12. DOCUMENT ANNOUNCEMENT (any limitation to the bibliographic announcement of this document. This will normally correspond to the Document Availabilty (11). However, where further distribution (beyond the audience specified in 11) is possible, a wider announcement audience may be selected.)

UNCLASSIFIED

SECURITY CLASSIFICATION OF FORM

13. ABSTRACT (a brief and factual summary of the document. It may also appear elsewhere in the body of the document itself. It is highly desirable that the abstract of classified documents be unclassified. Each paragraph of the abstract shall begin with an indication of the security classification of the information in the paragraph (unless the document itself is unclassified) represented as (S), (C), (R), or (U). It is not necessary to include here abstracts in both offical languages unless the text is bilingual).

(U) A Computer Simulation of a BPSK Direct Sequence Spread Spectrum System has been developed to investigate the effects of adaptively filtering stable tone interferers covering twenty per cent of the communication band. Preliminary results for the swept tone interfering signal are also presented and shows that the block filter is effective when the tones are stable, but degrades in performance as the tone sweeps quickly across the band. To combat the non-stationary signal environment, either shorter block lengths may be required or the use of recursive algorithms that are based on gradient or least square concepts may have to be examined.

14. KEYWORDS, DESCRIPTORS or IDENTIFIERS (technically meaningful terms or short phrases that characterize a document and could be helpful in cataloguing the document. They should be selected so that no security classification is required. Identifiers, such as equipment model designation, trade name, military project code name, geographic location may also be included. If possible keywords should be selected from a published thesaurus. e.g. Thesaurus of Engineering and Scientific Terms (TEST) and that thesaurus-identified. If it is not possible to select indexing terms which are Unclassified, the classification of each should be indicated as with the title.)

spread spectrum
spectrum estimation
computer simulation
autoregressive processes
maximum entropy method
least squares
adaptive filtering
signal excision
Weiner filtering
linear prediction